

VŠB – Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra informatiky

Absolvování individuální odborné praxe
Individual Professional Practise in the Company

VŠB - Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra informatiky

Zadání bakalářské práce

Student: **David Lesník**

Studijní program: B2647 Informační a komunikační technologie

Studijní obor: 2612R025 Informatika a výpočetní technika

Téma: **Absolvování individuální odborné praxe**
Individual Professional Practise in the Company

Zásady pro vypracování:

1. Student vykoná individuální praxi ve firmě: Poski.com s.r.o.
2. Struktura závěrečné zprávy:
 - a. Popis odborného zaměření firmy, u které student vykonal odbornou praxi a popis pracovního zařazení studenta.
 - b. Seznam úkolů zadaných studentovi v průběhu odborné praxe s vyjádřením jejich časové náročnosti.
 - c. Zvolený postup řešení zadaných úkolů.
 - d. Teoretické a praktické znalosti a dovednosti získané v průběhu studia uplatněné studentem v průběhu odborné praxe.
 - e. Znalosti či dovednosti scházející studentovi v průběhu odborné praxe.
 - f. Dosažené výsledky v průběhu odborné praxe a její celkové zhodnocení.

Seznam doporučené odborné literatury:

Podle pokynů konzultanta, který vede odbornou praxi studenta.

Formální náležitosti a rozsah bakalářské práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí bakalářské práce: **RNDr. Eliška Ochodková**

Konzultant bakalářské práce: Ing. Tomáš Posker

Datum zadání: 20.11.2009

Datum odevzdání: 07.05.2010



doc. Dr. Ing. Eduard Sojka
vedoucí katedry





prof. Ing. Ivo Vondrák, CSc.
děkan fakulty

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

V Ostravě dne

.....

Abstrakt

Cílem práce je popsat absolvování dvousemestrální individuální bakalářské praxe ve společnosti Poski.com s.r.o. na pozici webového vývojáře. Práce je zaměřena především na popis technologií užitých k tvorbě Internetových aplikací a vybraných projektů, které pomocí nich byly implementovány. Je zde shrnuto odborné zaměření firmy a pracovní zařazení studenta. Dále jsou popsány využití vědomosti, které student získal vysokoškolskou výukou a také chybějící vědomosti, jež byly nutné k úspěšnému dokončení projektů. Práce je shrnuta v podobě přínosů do života studenta.

Klíčová slova

webová aplikace, vývoj aplikace, programování, PHP, Subversion, Tapestry, framework, PoskiPHP, Symfony, MVC, šablonovací systém, Smarty, DWOO, Javascript, AJAX, jQuery, databáze, MySQL, ORM, Doctrine, YAML, dibi, Pohoda, HTML, CSS

Abstract

The goal of this thesis is to describe the completion of the two semesters of individual bachelor practise in the company Poski.com s.r.o. on job position web developer. Thesis is mostly focused on describing technologies used for developing Internet applications and choosen projects, which were developed by them. There is summed specialization of the company and job position of the student. Next, there is described used student's knowledge obtained by university education and also missing knowledge required to successfully finish projects. The thesis is summarized in the form of benefits to student's life.

Keywords

web application, application development, programming, PHP, Subversion, Tapestry, framework, PoskiPHP, Symfony, MVC, template engine, Smarty, DWOO, Javascript, AJAX, jQuery, database, MySQL, ORM, Doctrine, YAML, dibi, Pohoda, HTML, CSS

Seznam použitých symbolů a zkratek

AJAX – Asynchronous JavaScript and XML

CMS – Content Management System

CRM – Customer Relationship Management

CSS – Cascading Style Sheets

FTP – File Transfer Protocol

HTML – HyperText Markup Language

IDE – Integrated Development Environment

JS – JavaScript

MVC – Model-View-Controller

ORM – Object-Relational Mapping

PHP – Hypertext preprocesor

SEO – Search Engine Optimization

YAML – YAML Ain't Markup Language

Obsah

1	ÚVOD	1
2	ODBORNÉ ZAMĚŘENÍ FIRMY, PRACOVNÍ ZAŘAZENÍ	2
2.1	Firma Poski.com, s.r.o.	2
2.2	Pracovní zařazení studenta.....	2
3	ZADANÉ ÚKOLY	3
3.1	Programovací technologie	3
3.1.3	Tapestry	3
3.1.4	PHP	3
3.1.5	Šablonovací systémy	7
3.1.6	Javascript.....	8
3.2	HTML, CSS	9
3.3	MySQL.....	9
3.3.3	PoskiPHP.....	10
3.3.4	Doctrine.....	10
3.3.5	díbi.....	11
3.4	Zadané projekty	11
4	ŘEŠENÍ ÚKOLŮ	12
4.1	benefit4u.cz.....	12
4.2	spotrebice-whrilpool.cz.....	14
4.3	kvetiny-kvetinarstvi.cz.....	15
4.4	sundayphoto.cz.....	16
4.5	silektro.cz	16
5	ZÁVĚR.....	17
5.1	Uplatněné znalosti a dovednosti získané v průběhu studia.....	17
5.2	Chybějící znalosti a dovednosti	17
5.3	Zhodnocení praxe.....	17
6	SEZNAM OBRÁZKŮ	19
7	LITERATURA	20

1 Úvod

Práce pojednává o dvousemestrální bakalářské praxi absolvované ve společnosti Poski.com, s.r.o. na pozici webového vývojáře.

V první části je popsána odborná specializace firmy a pracovní zařazení mé osoby do pracovního procesu.

Druhá, největší část práce, je věnována technologiím, se kterými jsem pracoval a projektům pomocí nich vytvořených. Čtenář je seznámen s jazyky určenými pro programování webových aplikací, a to jak na straně serveru, tak na straně klienta a se vzorem pro návrh aplikace, který s rozšířením webového vývoje nabývá na stále větší popularitě. Další část kapitoly popíše výhody a nevýhody použití frameworků, zmíním se o dvou, se kterými jsem během praxe pracoval. Ve stejné kapitole popíši, jak byl vývoj aplikace zpřehledněn použitím šablonovacích systémů. U všech projektů bylo použito některé ORM, proto v této kapitole budou také popsány.

Další část práce předvádí použití popsaných technologií na vybraných projektech. Ty byly vybrány tak, aby demonstrovaly schopnost pracovat i například s ekonomickým systémem či bankovní aplikací pro platby platební kartou.

Poslední část práce shrnuje poznatky, které jsem nabyt na vysoké škole a využil při řešení úkolů, také podává přehled o chybějících znalostech. Závěrem shrnuji přínos praxe do svého života.

2 Odborné zaměření firmy, pracovní zařazení

2.1 Firma Poski.com, s.r.o.

Odbornou praxi jsem vykonával ve společnosti Poski.com s.r.o. se sídlem na Sokolské třídě 871/6 v centru Ostravy.

Činnost firmy můžeme rozdělit do několika hlavních kategorií [1].

Zřejmě největší z nich je tvorba vlastních Internetových aplikací, patří zde například Internetový obchod ve třech variantách, systém pro správu obsahu webových stránek (CMS), dále pak informační systém určený výhradně pro realitní kanceláře (Poski REAL) a aplikace přizpůsobená ke správě městských či obecních stránek (WEBObce). Na zákazníkovo přání je možné realizovat i systém pro řízení vztahů se zákazníky (CRM).

Další velkou oblastí, již se firma zabývá, je grafický návrh, tvorba, případně modernizace webových prezentací pro jednotlivce či firmy.

Společnost Poski.com je také schopna pomocí postupů a nástrojů přivést na stránky cílovou skupinu uživatelů, pro které je webová prezentace určená. Tento proces se nazývá SEO.

2.2 Pracovní zařazení studenta

Po úspěšném absolvování vstupního pohovoru jsem, dle smlouvy, nastoupil na pracovní pozici pomocný programátor PHP a MySQL. Po osvojení si firemního PHP Frameworku jsem byl na některé projekty nasazen jako hlavní programátor.

Jestliže se zaměříme na povahu konkrétních úkolů, týkaly se především Internetového obchodu. Mezi použitá vývojová prostředí patří Aptana, Eclipse a NetBeans.

Po celou dobu praxe mne měl na starosti výrobní ředitel společnosti Ing. Tomáš Posker.

3 Zadané úkoly

Před řešením problémů z praxe bylo nutné si osvojit různé vývojové technologie a nabýt požadované znalosti. Zde představím technologie, jejichž nastudování bylo mým úkolem, poté obecně popíši zadání projektů.

3.1 Programovací technologie

3.1.3 Tapestry

Jednou z prvních technologií, se kterými jsem se během praxe seznámil, byl framework Tapestry[2] pro vývoj webových aplikací založených na programovacím jazyku Java. Využit měl být pro vývoj rozsáhlého portálu věnovaného rekreačním místům v několika evropských zemích. Po projití několika tutoriálů jsem usoudil, že je poměrně složitý, a to i přesto, že byl pro Javu, kterou docela zvládám. Jen rozběhnutí první webové aplikace s ORM Hibernate mi zabralo asi dva dny. Naštěstí pro mě z jeho použití sešlo (chabá podpora pro SEO) a bylo nahrazeno jazykem PHP.

3.1.4 PHP

Vývoj v jazyce PHP tvořil mou hlavní pracovní náplň. Jedná se o jednu z nejpoužívanějších technologií pro vývoj webových aplikací na straně serveru [3]. Svou popularitu získal hlavně díky své jednoduchosti, flexibilitě a snadným propojením s HTML kódem. Od páté verze je možné při programování používat objektově orientovaný přístup. To sebou nese výhodu psaní přehlednějšího kódu a zlepšení jeho údržby. Dřívější procedurální přístup je možné samozřejmě používat i nadále.

Frameworky

Pro vývoj v PHP se v poslední době čím dál více používají frameworky. Jsou to v podstatě softwarové struktury[4], jejichž cílem je usnadnit programátorovi psaní aplikace, aby se mohl soustředit na skutečně tvůrčí činnost a nemusel se přitom zabývat základními problémy. Velkou výhodou je také to, že programátora obvykle vedou k použití správných technik a mnohdy za něj ošetří případné bezpečnostní díry.

Z nevýhod můžeme jmenovat snížení rychlosti zpracování kódu a také nutnost vyhradit si čas pro jeho nastudování.

Velkou skupinu tvoří open source frameworky, z nejpoužívanějších v ČR můžeme jmenovat například Nette, Zend, Symfony, CakePHP nebo CodeIgniter. Všechny jmenované mají širokou komunitu vývojářů pracujících s nimi, čímž je zajištěna podpora, neví-li si programátor rady.

Další skupinou frameworků jsou ty, které vznikly pro nějaký specifický účel. Může jím být například snaha programátora napsat si vlastní pracovní rámec nebo, v případě firmy, urychlit plnění stále se opakujících požadavků klientů.

V porovnání s open source frameworky spočívá jejich výhoda v tom, že tvůrci mají dokonalé znalosti jeho fungování a je přizpůsoben jejich potřebám. Nevýhodou často bývá nedostatečná dokumentace. Tento fakt je klíčový při přijímání nových spolupracovníků, kteří se nemají při tvorbě aplikace o co opřít, případně neznají možnosti, které jim framework nabízí. Další nevýhodou může být vyšší náchylnost k chybám; jednotlivci, případně malá skupina nemusí dosahovat takových znalostí jako vývojáři open source frameworků.

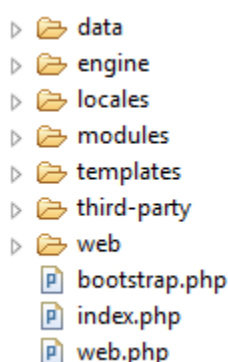
Během praxe jsem se seznámil s oběma typy frameworků. První z nich byl firemní PoskiPHP, druhým bylo Symfony. Oba v krátkosti představím.

3.1.4.1 PoskiPHP

V úvodu své praxe jsem byl seznámen s firemním PHP Frameworkem PoskiPHP, se kterým jsem pracoval po celou praxi. Hlavním vývojářem je Jakub Macek.

Popis adresářové struktury projektu (obrázek 1)

- data – slouží k ukládání obrázků, souborů a dočasných souborů
- engine – třídy tvořící jádro frameworku
- locales – soubory s jazykovými mutacemi
- modules – administrační moduly
- templates – soubory šablon (Smarty, DWOO, obě viz dále)
- third-party – knihovny třetích stran (TinyMCE, šablonovací systém, AJAXové knihovny)
- web – layout stránky, JS a CSS soubory
- soubory, ve kterých se provádí různá globální nastavení aplikace.



Obrázek 1: Adresářová struktura frameworku PoskiPHP

Framework také nabízí vestavěné ORM (viz dále).

Cílem frameworku je co nejvíce usnadnit vytvoření administračního rozhraní pro datové entity, které jsou požadovány zákazníkem. Pro každou entitu je vytvořen jeden modul, jejich registrace pak probíhá v souboru index.php.

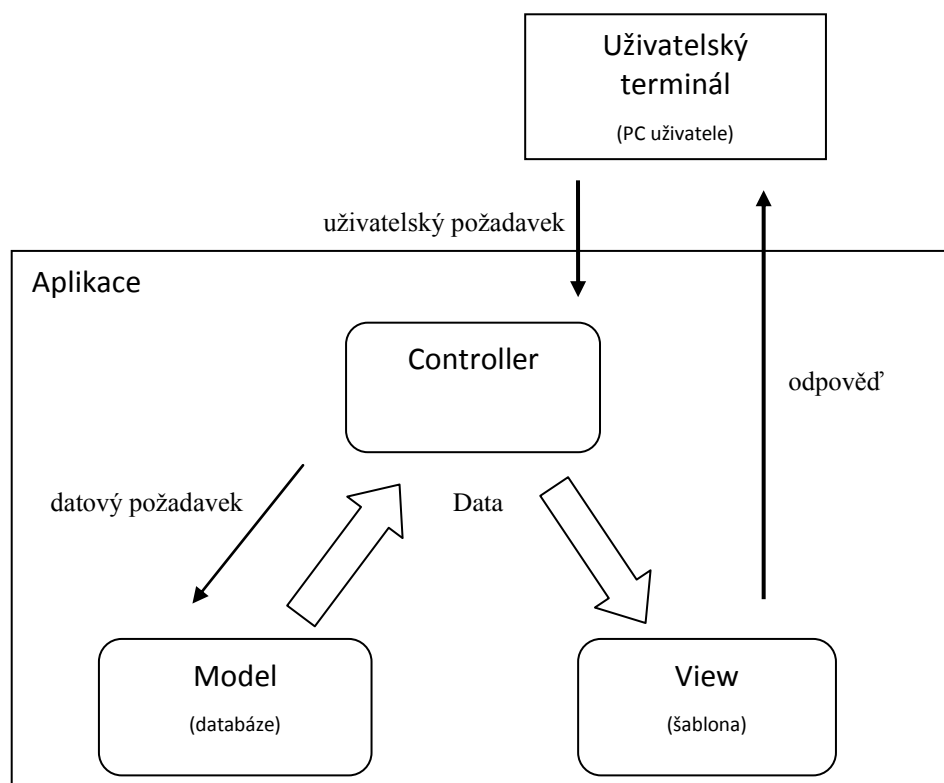
Výhody frameworku jsou zřejmé, fungující administrační část webové aplikace jsme schopni vytvořit během pár chvil. Automaticky se nám vytváří tabulky v databázi, definici položek provádíme přímo v modulovém souboru.

Nevýhody mohu zhodnotit z pohledu člověka, který začal s frameworkem pracovat až po té co byla největší část hotova. Největší spatřuji v chybějící dokumentaci, která byla v době, kdy jsem začínal ve společnosti Poski.com praxi, nulová, kód nebyl a není okomentovaný. Na začátek mne tedy čekal poměrně velký šok. Trávil jsem dlouhé hodiny hledáním postupů jak vyřešit základní úkony. V současnosti je snaha vytvářet firemní wiki. Jako další bych viděl poměrně nepřehledný výpis chyb ve vykonávání kódu, někdy bylo opravdu složité najít jejich původ.

3.1.4.2 Symfony

Symfony je open source MVC framework pro PHP [5].

MVC



Obrázek 2: Architektura MVC

MVC (Model-View-Controller) je architektonický vzor [6] pro přístup k návrhu aplikací. Jeho smyslem je oddělení souvisejících funkčních celků aplikace do skupin podle toho co je jejich úkolem. Těmto skupinám také přiřazuje rozsah úkolů, které mají řešit. Vytváří tak jasnou strukturu aplikace, jejíž následná údržba je mnohem snadnější v porovnání s jinou technikou návrhu.

Autorem definice MVC je z roku 1979 norský profesor Trygve Reenskaug[7], od té doby přístup prodělal několik změn.

Rozdělení aplikace do tří částí, obrázek 2:

Model – business logika aplikace, včetně databáze

View – část aplikace, která je prezentována uživateli (také pohled, šablona)

Controller – slouží k načtení dat z modelu a jejich odeslání na správnou šablonu

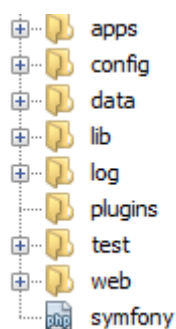
Symfony je svým zaměřením určeno pro podnikovou (enterprise) sféru. V počátcích jeho vývoje za ním stál francouzský programátor Fabien Potencier, poté založil konzultační firmu Sensio Labs, která jeho vývoj nadále zastřešuje.

Klíčové vlastnosti frameworku:

- založen na MVC architektuře
- ovládání pomocí příkazové řádky
- integrace s ORM – Doctrine (viz dále), Propel
- integrace s JS knihovnami – jQuery, Prototype
- generování administrace
- statistiky pro vývoj – developer toolbar
- konfigurační soubory ve formátu YAML[8] (standardně)
- mnoho doplňků
- bohatá dokumentace

Projekt můžeme vytvořit buď manuálně nakopírováním zdrojových souborů, nebo použít doporučený postup pomocí příkazové řádky.

Jak vypadá struktura nově vytvořeného projektu, ukazuje obrázek 3.



Obrázek 3: Adresářová struktura frameworku Symfony

Obecně se projekt skládá z aplikací (apps), obvykle to bývá uživatelská část (front-end) a administrační část (back-end). Aplikace se dále dělí do modulů, které obsahují jednotlivé akce a šablony.

Z významných webových projektů a společností, které Symfony používají, můžeme jmenovat například Yahoo.com, delicious.com, z českých pak například společnost Symbio.

V současnosti je posledním stabilní verzí 1.4, koncem roku 2010 je naplánováno vydání verze 2.0, která by měla přinést mnohonásobné zrychlení oproti ostatní frameworkům, změny v architektuře a přehlednější ladící informace [9].

3.1.5 Šablonovací systémy

Šablonovací systém je technologie pro oddělení aplikačního a prezentačního kódu aplikace, napomáhá tak k větší přehlednosti projektu a také usnadňuje delegování práce v týmu, kde jsou programátoři a kódeři rozděleni.

Za svou praxi jsem poznal oba šablonovací systémy, se kterými firemní PoskiPHP framework umí pracovat.

3.1.5.1 Smarty

Smarty je šablonovací systém pro PHP [10]. Jeho silnou zbraní je možnost ukládat výsledné šablony do vyrovnávací paměti a tím celou aplikaci výrazně zrychlit. Jsou dva způsoby jak toho docílit, šablona se uloží do vyrovnávací pouze zkompilovaná, při jejím každém spuštění se musí vykonat PHP kód, nebo se uloží jako HTML soubor, který se pak posílá do prohlížeče uživatele [11].

Smarty využívá speciální syntaxi příkazů. K proměnným poslaným do šablony

```
$smarty->assign('promenna', 'jeji hodnota');
```

se přistupuje: `{$smarty}`

A na stránce se vypíše: `jeji hodnota`

Smarty umožňuje i příkazy běžné pro jiné programovací jazyky, do šablony lze tedy zavést určitou logiku.

3.1.5.2 DWOO

DWOO tvoří novější alternativu pro Smarty, je psáno s využitím objektového programování [12]. Nabízí rychlejší zpracování kódu a vylepšené funkce. Jeho syntaxe je také přívětivější, neboť je bližší klasickému programovacímu jazyku.

Výhodou obou šablonovacích technologií je fakt, že s využitím vyrovnávací paměti jsou rychlejší než standardně vykonávaný kód. Nevýhodou je skutečnost, že pro jejich použití je nutné se naučit, ne zrovna jednoduchou, syntaxi.

3.1.6 Javascript

Javascript je zcela objektově orientovaným jazykem [13], který je používán především pro programování webových stránek na straně klienta, tedy přímo v prohlížeči. Po svém uvedení v roce 1995 byl jazyk dlouhá léta opomíjen, prý kvůli chybám, bezpečnosti, a především kvůli chabé podpoře tehdejších prohlížečů. Nicméně, jak šel čas, výrobci podporu javascriptu přidávali a také programátoři se jej učili. Velký skok v jeho popularitě znamenal pojem AJAX (viz dále). V dnešní době je javascript respektovaným jazykem a tvoří nedílnou součást webových stránek.

Nevtíravý Javascript

Tento pojem vyjadřuje techniku pro vývoj moderních webových stránek na straně klienta, kdy uživateli nabídneme pokročilý obsah, pokud jej podporují, v opačném případě zobrazíme základní, zato však plně funkční obsah stránky [14].

AJAX

AJAX je soubor technologií, které umožňují načíst nová data na stránku, aniž by uživatel musel provádět její aktualizaci. Toho je dosaženo pomocí objektu XMLHttpRequest (pro prohlížeče jiných výrobců než Microsoft) nebo ActiveXObject (v případě prohlížeče od společnosti Microsoft).

Vytvoření XMLHttpRequest objektu:

```
var xmlhttp;  
if (window.XMLHttpRequest) {  
    xmlhttp = new window.XMLHttpRequest();  
} else if (window.ActiveXObject) {  
    xmlhttp = new ActiveXObject("Microsoft.XMLHTTP");  
}
```

Použitím AJAXu odpadá nutnost načítat celou stránku, aby se aktualizovala pouze část jejího obsahu. Tímto přístupem se zbytečně neplýtvá s přenosovou kapacitou sítě a práce je plynulejší.

Nevýhodou může být jeho nevhodné použití, neboť s novými daty se mění stav aplikace, ale nemění se URL adresa, uživatel tedy může být zmaten.

3.1.6.1 jQuery

jQuery je Javascriptová knihovna [15] sloužící ke zjednodušení práce vývojářů pracujících na zlepšení uživatelské přívětivosti stránky. Knihovna je navržena tak, aby minimalizovala psaní zbytečně dlouhého kódu a pomocí volání vestavěných funkcí prováděla komplexní operace.

Knihovna může být dále rozšířena o celou řadu doplňků.

Za jejím vznikem stojí Javascriptový guru John Resig [16].

Základní použití

S nevtíravým Javascriptem často potřebujeme vybrat určitou skupinu elementů a přiřadit jim určité události na akce provedené uživatelem.

Následující kód přidá po přejetí kurzorem myši elementu s CSS třídou `styl` třídu `dalsistyl`, po posunu kurzoru mimo element se třída `dalsistyl` odebere.

```
$(document).ready(function(){
    $(".styl").mouseover(function(){
        $(this).addClass("dalsistyl");
    }).mouseout(function(){
        $(this).removeClass("dalsistyl");
    });
});
```

Můžeme si všimnout, že k vybrání elementů se používá stejná syntaxe jako při práci s vizuálními styly – CSS. Znak `$` značí objekt jQuery.

3.2 HTML, CSS

I když práce s těmito technologiemi nebyla mou hlavní pracovní náplní, mnohokrát jsem jejich znalost potřeboval.

3.3 MySQL

MySQL byl jediný systém řízení báze dat, se kterým jsem na praxi pracoval. Používá několik typů úložišť [17]. Nejznámějším a nejrozšířenějším je MyISAM, který nabízí vysoký výkon, ale neumí pracovat s cizími klíči a transakcemi. Dalším velmi známým typem úložiště je InnoDB, který odstraňuje nedostatky MyISAM, ale jeho výkon je nízký.

Jelikož je pro vývoj webu velmi podstatná rychlost přístupu k datům a mnohem častěji dochází k výběru dat, než jejich přidávání, či aktualizaci, bylo použito úložiště MyISAM.

ORM

Výhodou tohoto přístupu práce s databází je, že programátor nemusí psát čisté SQL dotazy do kódu aplikace, tím se minimalizuje chyba zvaná SQL injection (podvrhnutí řetězce, kterým se změnil dotaz) a také je kód čistější. Nevýhodou bývá zpomalení.

3.3.3 PoskiPHP

Součástí firemního frameworku bylo i ORM, které mi velmi usnadňovalo práci. Následuje ukázka práce s ním:

```
<?php $foo = o('module_name')->load(Fand(Feq('title', 'název'),  
Feq('display_HP', 1)), true); ?>
```

Příkaz přiřadí `foo` první objekt (`true` na konci), který vyhovuje podmínce, že `title` je roven slovu `název` a zároveň má nastaven atribut `display_HP` na 1.

Klasické SQL dotazy je samozřejmě možné psát také, a i zde jsou případné útoky ošetřeny, děje se tak pomocí parametrů.

3.3.4 Doctrine

Doctrine je jeden z nejznámějších open source ORM pro PHP [18]. Během praxe jsme jej použili v kombinaci s frameworkem Symfony.

Příklad vytvoření tabulky se třemi sloupci – id, jméno, příjmení:

```
Clovek:  
columns:  
    jmeno as jmeno:  
        type: string(255)  
    prijmeni as prijmeni:  
        type: string(255)
```

Můžeme si všimnout, že není definován atribut `id`, ten si Doctrine vytvoří sám.

Vložení záznamu a jeho výběr můžeme provést takto:

```
$clovek = new Clovek();  
$clovek->jmeno = "David";  
$clovek->prijmeni = "Lesnik";  
$clovek->save(); //ulozeni  
  
$clovek = Doctrine::getTable('Clovek')->findOneById(1); //vybrani
```

Jelikož je to první záznam, má `id` 1; pro ukázkou si to můžeme dovolit.

3.3.5 dibi

Na projektu, u kterého bylo užito Doctrine byla využita, avšak v menší míře, i databázová vrstva (layer) s názvem dibi, autorem je známý český PHP programátor David Grudl [19].

Ukázka výběru řádku:

```
$clovek = dibi::select('*')->from('Clovek')->where(id=%i', 1)->fetch();
```

3.4 Zadané projekty

Hlavní pracovní náplní nebylo jen studium technologií, ale především jejich využití v projektech. Většinou se jednalo o projekty nové, byly však mezi mini i požadavky na úpravu stávajících webů.

Obvykle jsem současně pracoval na více projektech současně. Ze začátku jsem byl veden hlavním programátorem, poté jsem řešil projekty sám, někdy s konzultací jednotlivých úkonů a řešení.

Seznam nových projektů, v závorce uvádím dobu trvání

alfa-com.cz (2 měsíce), eshop.barvy-laky-fasady.cz (1.5 měsíce), benefit4u.cz (4 měsíce), charis.cz (měsíc), haraldinterier.cz (2 měsíce), iglootechnika.cz (3 měsíce), interdecor.cz (14 dnů), jurotrans.cz (14 dnů), kerastol.cz (2 měsíce), kvetiny-kvetinarstvi.cz (3 měsíce), nabytekdokoupelny.cz (2.5 měsíce), silektro.cz (4 měsíce), spotrebice-whirlpool.cz (2 měsíce), sundayphoto.cz (měsíc), zlatobezdph.cz (měsíc)

Některé z vyjmenovaných projektů jsou hotové, na některých stále pracuji.

Ke každému novému projektu bylo nutné připravit administrační rozhraní, definovat požadované položky, upravit systém podle požadavků, připravit a napojit šablony na HTML stránku.

4 Řešení úkolů

Za celou svou praxi jsem pracoval na mnoha projektech, zde popíšu ty, které považuji za nejzajímavější.

4.1 benefit4u.cz

benefit4u.cz je nejrozsáhlejší projekt, na kterém jsem za svou praxi pracoval jako jediný vývojář.

Jedná se o internetový obchod, jehož účelem není jen prodávání produktů, ale komplexní správa obchodních zástupců, partnerů a zákazníků, to vše s propojením na ekonomický informační systém.

Základní charakteristika:

- internetový obchod
 - o produkty
 - o skupiny produktů
 - možnost provádět hromadné snižování cen produktů
 - nastavení ceny za dopravu
 - o výrobci
 - o zákazníci – rozdělení na:
 - registrované
 - zaměstnanci partnera
 - neregistrované
 - každá skupina má jinou úroveň ceny, za kterou nakupuje
 - o objednávky
 - o správa recyklačních poplatků
- propojení s ekonomickým systémem Pohoda
 - o objednávky
 - o správa produktů
- správa obchodních zástupců
 - o zavést do systému přístupové role
 - o statistický modul
- správa firem – Partnerů
 - o správa zaměstnanců jednotlivých partnerů
 - generování přihlašovacích hesel pro zadaný počet zákazníků
 - o export seznamu zaměstnanců s kódy do souboru Excelu
- záznam komunikace s firmami
 - o přeposílání hlavnímu administrátorovi
 - o také princip upozornění na události pomocí emailu
- správa reklamních bannerů a aktualit

Projekt je postaven na firemním PoskiPHP frameworku, jelikož spousta požadovaných funkcí není v základu podporována, bylo nutné je doprogramovat.

Recyklační poplatky a cena za dopravu:

Obě položky jsou vyplňovány ve skupinách produktů, k ceně produktů se přičítají po vložení do košíku. V obou případech bylo nutné upravit framework přidáním položek a změnit počítání ceny produktu.

Slevy produktů

U každého produktu je možné zadat vlastní procentuální slevu, o kterou se má jeho cena snížit. Cenu je možné snižovat i pro celou skupinu produktů. Snižuje se vždy z ceny základní, nikoliv navýšené o recyklační poplatek a dopravu. Každá skupina zákazníků může mít různou slevu.

Generování uživatelů:

Bylo nutné umožnit generování zadaného počtu uživatelských účtů – registrovaných zákazníků.

Pro konkrétního partnera uložím do tabulky se zákazníky počet požadovaných účtů. Platí, že přihlašovací jméno si zvolil partner a heslo je generováno náhodně. Před vložení nového záznamu musím zkontrolovat neexistenci duplicitního hesla a jména.

Tyto účty následně uložím do souboru, poté načtu všechny účty pod daným partnerem a ty uložím do souboru na další list. Tento soubor pak uložím ke konkrétnímu partnerovi, který si jej pak může stáhnout ve svém profilu.

Pro řešení práce se souborem ve formátu Excelu jsem použil knihovnu PHPEXcel [20], která umožňuje velmi pokročilou práci se soubory tohoto typu.

Propojení s ekonomickým systémem Pohoda

Pracovalo se s objednávkami a produkty pomocí externího programu. Jeho autorem je Vítězslav Prášil.

Bylo nutné nastudovat formát, se kterým Pohoda umí pracovat, poté do něj stačilo dosadit data z jednotlivých položek, uložit a přenést výsledek.

Práce s Pohodou je velmi komplexní téma, které zde nejsem schopný popsat.

Produktový filtr

Ve výpisech produktů jsem zavedl možnost filtrovat aktuálně zobrazené položky podle kategorií. Vyžadovalo to upravit systém a externí knihovnu, pomocí které je prováděno stránkování.

Systém

Byly provedeny změny co se týkájí se přístupu k určitým položkám.

JavaScript

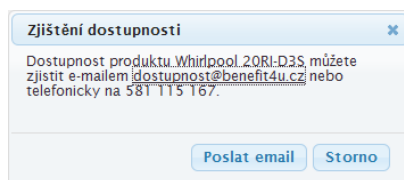
Zavedl jsem možnost řazení výpisu tabulek pomocí jQuery pluginu TableSorter 2.0 [21]. Tento plugin umožňuje řazení bez nutnosti opětovného načtení stránky. Co jsem potřeboval změnit, byla struktura tabulky, aby s ní plugin uměl pracovat.

Aktivaci řazení pak po importování souboru s pluginem a s knihovnou jQuery provedeme:

```
$(document).ready(function(){  
    $(".list").tablesorter();  
});
```

kde `list` je CSS třída řazené tabulky.

Další implementovanou věcí s využitím jQuery bylo dialogové okno [22] pro zjištění dostupnosti. Do předmětu v emailu se automaticky vyplní kód produktu.



Obrázek 4: Dialog pro zjištění dostupnosti produktu

benefit4u.cz byl zřejmě nejnáročnějším projektem, který jsem během praxe tvořil, trval několik měsíců. Naučil a vyzkoušel jsem si na něm věci, o kterých jsem dříve ani neslyšel. Projekt je plně funkční a běží na příslušné doméně.

4.2 spotrebice-whirlpool.cz

Projekt je svými požadavky podobný předešlému uvedenému, nabízí však navíc možnost platby kartou, splátkový prodej a volbu prodloužené záruky u jednotlivých produktů. Její volbu je možné provádět jak v detailu, tak v košíku. Nebylo zde nutné implementovat přístupové role a věci spojené s uživatelskými účty.

Je specifický tím, že produktový katalog je shodný s katalogem mateřského webu whirlpool.cz. Bohužel pro mě, struktura skupin je poněkud jiná bylo tedy nutné přijít s řešením.

Některé produkty byly také importovány z katalogů ve formátu souboru Excel.

Projekt je stále ve fázi vývoje.

4.3 kvetiny-kvetinarstvi.cz

Jedná se o realizaci prodeje květin s možností platby platební kartou.

Základní charakteristika:

- Internetový obchod
 - o produkty
 - každý produkt je nabízen ve třech velikostech
 - každá velikost má jinou cenu
 - o skupiny produktů
 - o zákazníci
 - o objednávky
- platba objednávky pomocí platební karty
 - o komunikace s platebním systémem banky
 - o zamezení podvodu

Platby kartou

Pro online placení bylo nutné využít zabezpečené připojení pomocí SSL HTTPS.

Postup vyřízení a zaplacení objednávky:

1. zvolení on-line platby
2. přesměrování na formulář s výběrem platební karty
3. odeslání informací na systém banky
4. kontrola odeslaných informací bankou
5. uživatel vyplní údaje o své platební kartě
6. autorizace platebním systémem banky
7. opětovná kontrola údajů
8. transakce je úspěšná

JavaScript

Také u toho projektu jsem využil jeden z pluginů pro jQuery, konkrétně kalendář [23] pro pomoc s vyplněním data doručení objednávky.



Obrázek 5: Interaktivní kalendář

kvetiny-kvetinarstvi.cz je projekt, u kterého jsem opět poznal něco, co jsem doposud neznal. Bohužel produktový katalog se stále plní, není možné se tedy na projekt podívat živě.

4.4 sundayphoto.cz

sundayphoto.cz je projekt, na kterém v současnosti pracuji ve spolupráci s hlavním firemním vývojářem Jakubem Mackem. Jedná se o Internetovou fotobanku, jejíž fotografie je možné zakoupit. K dispozici je několik jazykových mutací. Tento projekt je vyvíjen pomocí PHP frameworku Symfony a ORM Doctrine.

Mými prozatímní úkoly byly vytvořit a obsloužit několik formulářů. U registračního je pomocí AJAXu kontrolována duplicita jména, jsou kontrolovány i některé další údaje, například shoda hesel.

Na projektu si cením toho, že jsem se setkal s frameworkem Symfony, který patří mezi nejlepší PHP frameworky.

4.5 silektro.cz

Jde o další z projektů, tentokrát s nasazením systému pro správu obsahu webových stránek.

Základní charakteristika:

- několik modulů
 - o každý pro správu jednotlivých částí
- přihlášení dealerů
- ceníky
 - o několik skupin ceníků
 - o každý dealer je v nějaké ceníkové skupině

I v tomto projektu jsem si vyzkoušel dovednosti s Javascriptem. Například v menu, kde nejdříve vypíši všechny položky, poté je za pomoci Javascriptu skryji. Dalším prvkem může být zlepšení administrace, kdy se skryjí položky, které spolu nesouvisí.

5 Závěr

5.1 Uplatněné znalosti a dovednosti získané v průběhu studia

Ze získaných znalostí v průběhu studia jsem si během praxe nejvíce cenil znalostí týkajících se databází, především jejich návrhu. Nebyl problém navrhnout databázové schéma jen při čtení požadavků zákazníka. Kladl-li by se větší důraz na databázovou vrstvu projektů a nebyla-li použita pouze základní MySQL databáze, byl bych schopen navrhnout i odpovídající výkonové vylepšení, především použití indexů a jiných typů tabulek. Další věcí, které si cením, je analytické myšlení, které jsem při studiu vylepšil. Byl jsem schopen navrhnout systém rychleji i s promyšlenými detaily, ne se všemi samozřejmě. Studium mě také připravilo výukou několika programovacích jazyků.

5.2 Chybějící znalosti a dovednosti

V průběhu studia jsem získal mnoho nových znalostí a dovedností. Nicméně pro práci webového vývojáře jsem se musel potřebnými znalostmi dovybavit především samostudiem. Ze školy by se hodila větší praxe v SQL příkazech, které jsme si psali na papír (to už se tuším změnilo) a také hlubší znalost návrhových vzorů, především s příklady jejich praktického užití ve vývoji.

Jelikož je v současné době oblast informačních technologií velmi rozsáhlé téma, není možné po bakalářském studijním programu v tomto oboru požadovat vysokou úroveň specializace, ale spíše všeobecný přehled technologií a teoretických znalostí. Domnívám se, že to škola splnila. Pro specializaci je tak nejdůležitější samostudium a aktivita jednotlivce.

5.3 Zhodnocení praxe

Během vykonávání praxe jsem získal mnoho nových znalostí a zkušeností. Většina z nich se týká vývoje webových aplikací.

Vyzkoušel jsem si práci s jedním z nejlepších frameworků, práci s verzovacím systémem Subversion, několika ORM. Dále jsem se naučil pracovat s firemním PHP frameworkem, zde bych však spíše vyzdvihl schopnost porozumět kódu, který psal někdo jiný a také trpělivost. Poznal jsem několik nových technologií, především šablonovací systémy. Získal jsem velmi dobré znalosti pro práci s Javascriptem a s knihovnou jQuery. Dále jsem zjistil, že jsem na tom s algoritmickým myšlením lépe než kdy dříve.

Praxe pro mne byla také osobnostním přínosem, poznal jsem několik nových a zajímavých lidí, měl jsem možnost účastnit se jednání s klienty, zlepšil jsem své komunikační schopnosti a zvýšilo se mi sebevědomí.

Byly však chvíle, kdy se mi ne tak úplně dařilo, bylo to zejména období na začátku praxe, kdy jsem se musel učit něco, k čemu vlastně není nikde napsán žádný text a co využiji jen v této firmě (firemní framework), dále mi byl k programování přidělen notebook, jehož výkon byl velmi nízký. Do toho všeho jsem musel plnit školní úkoly.

Celkově si velmi cením toho, že mi škola umožnila vykonat bakalářskou práci tímto způsobem., předvedla mi tak jeden z možných směrů pokračování mého života.

6 Seznam obrázků

Obrázek 1: Adresářová struktura frameworku PoskiPHP.....	4
Obrázek 2: Architektura MVC	5
Obrázek 3: Adresářová struktura frameworku Symfony.....	6
Obrázek 4: Dialog pro zjištění dostupnosti produktu	14
Obrázek 5: Interaktivní kalendář	15

7 Literatura

- [1] *Řešení | webdesign, redesign, údržba webových stránek, SEO, SEM, e-shop, CMS, CRM | Poski.com* [online]. 2009 [cit. 2010-04-29]. Dostupné z WWW: <<http://www.poski.com/reseni>>.
- [2] *Apache Tapestry - Apache Tapestry 5* [online]. 2009-05-08 [cit. 2010-04-30]. Dostupné z WWW: <<http://tapestry.apache.org/tapestry5.1/>>.
- [3] *PHP: Hypertext Preprocessor* [online]. 2010-04-20 [cit. 2010-05-01]. PHP: Hypertext Preprocessor. Dostupné z WWW: <<http://php.net/>>.
- [4] *Wikipedie, otevřená encyklopedie* [online]. 2006-05-22 [cit. 2010-05-02]. Framework - Wikipedie, otevřená encyklopedie. Dostupné z WWW: <<http://cs.wikipedia.org/wiki/Framework>>.
- [5] *Symfony | Web PHP Framework* [online]. 2010-05-02 [cit. 2010-05-02]. Dostupné z WWW: <<http://www.symfony-project.org>>.
- [6] Alternativy k MVC a závěrečné poznámky. *Zdroják - tvorba webových stránek a aplikací* [online]. 2009-05-15, [cit. 2010-05-02]. Dostupný z WWW: <<http://zdrojak.root.cz/clanky/alternativy-k-mvc-a-zaverecne-poznamky/>>.
- [7] REENSKAUG, Trygve The original MVC reports. In *The original MVC reports*. Oslo : University of Oslo, 2007-02-12 [cit. 2010-05-02]. Dostupné z WWW: <http://heim.ifi.uio.no/~trygver/2007/MVC_Originals.pdf>.
- [8] *The Official YAML Web Site* [online]. 2009-10-01 [cit. 2010-05-02]. Dostupné z WWW: <<http://www.yaml.org/>>.
- [9] *Symfony Reloaded - 2.0 Preview Release* [online]. 2010-02-13 [cit. 2010-04-11]. Dostupné z WWW: <<http://symfony-reloaded.org/>>.
- [10] *Smarty : Template Engine* [online]. 2009-10-01 [cit. 2010-05-02]. Dostupné z WWW: <<http://www.smarty.net/>>.
- [11] KAVALEK, Tomáš . Smarty, šablonovací systém - 1 (úvod). *AbcLinuxu.cz - Linux na stříbrném podnose* [online]. 2006-11-07, 1, [cit. 2010-05-02]. Dostupný z WWW: <<http://www.abclinuxu.cz/clanky/navody/smarty-sablonovaci-system-1-uvod>>.
- [12] *Home - Dwoo - a PHP Template Engine* [online]. 2009-08-27 [cit. 2010-05-02]. Dostupné z WWW: <<http://dwoo.org/>>.
- [13] RESIG, John. *Javascript a Ajax : Moderní programování webových aplikací*. první. Brno : Computer Press, a.s., 2007. Základy objektově orientovaného kódu, s. 41.

- [14] RESIG, John. *Javascript a Ajax : Moderní programování webových aplikací*. první. Brno : Computer Press, a.s., 2007. Nevětřivé skriptování DOM, s. 17.
- [15] *JQuery: The Write Less, Do More, JavaScript Library* [online]. 2010-03-27 [cit. 2010-05-02]. Dostupné z WWW: <<http://jquery.com/>>.
- [16] *John Resig - JavaScript Programmer* [online]. 2010-04-03 [cit. 2010-05-02]. Dostupné z WWW: <<http://ejohn.org/>>.
- [17] Chapter 13. Storage Engines. *MySQL :: Developer Zone* [online]. 2009-07-11, [cit. 2010-05-02]. Dostupný z WWW: <<http://dev.mysql.com/doc/refman/5.1/en/storage-engines.html>>.
- [18] *Doctrine - PHP Object Relational Mapper* [online]. 2010-04-26 [cit. 2010-05-02]. Dostupné z WWW: <<http://www.doctrine-project.org/>>.
- [19] GRUDL, David. Dibi - pokrokový databázový layer. *LaTrine* [online]. 2006-05-26, [cit. 2010-05-02]. Dostupný z WWW: <<http://latrine.dgx.cz/dibi-pokrokovy-databazovy-layer>>.
- [20] *PHPExcel* [online]. 2010-03-16 [cit. 2010-05-02]. Dostupné z WWW: <<http://phpexcel.codeplex.com/>>.
- [21] *JQuery plugin: Tablesorter 2.0* [online]. 2010-01-03 [cit. 2010-05-02]. Dostupné z WWW: <<http://tablesorter.com/docs/>>.
- [22] *JQuery UI - Home* [online]. 2009-07-07 [cit. 2010-05-02]. JQuery UI - Demos & Documentation. Dostupné z WWW: <<http://jqueryui.com/demos/dialog/>>.
- [23] *JQuery UI - Home* [online]. 2009-07-07 [cit. 2010-05-02]. JQuery UI - Demos & Documentation. Dostupné z WWW: <<http://jqueryui.com/demos/datepicker/>>.